2

ARPA/ONR Quarterly Report

CONTRACT NUMBER:	N00014-91-J-1985
CONTRACTOR:	Duke University
ARPA ORDER NO:	4331714-01/4-6-88
PROGRAM NAME:	N00014
CONTRACT AMOUNT:	$696,899.00
EFFECTIVE DATE OF CONTRACT:	July 1, 1991
EXPIRATION DATE OF CONTRACT:	June 30, 1994
PRINCIPAL INVESTIGATOR:	John H. Reif
TELEPHONE NUMBER:	(919) 660-6568
SHORT TITLE OF WORK:	Implementation of Parallel Algorithms
REPORTING PERIOD:	1 Apr 93 — 30 Jun 1993

The views and conclusions contained in this document are those of the author(s) and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Advanced Research Projects Agency or the U.S. Government.

DESCRIPTION OF PROGRESS

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. We are pleased to announce in particular two significant events.

First, Reif has recently had two books published on parallel algorithms and implementations for which he was editor -- "Synthesis of Parallel Algorithms" and "Parallel Algorithm Derivation and Program Transformation" (co-edited with R. Paige and R. Wachter).

Secondly, Peter Su, a graduate student from Dartmouth who moved to Duke to work on his Ph.D. on parallel algorithm implementations with Reif, defended his dissertation at Dartmouth in June 1993. Su's work at Duke was supported under this grant. Su's dissertation, "Efficient parallel algorithms for closest point problems", develops fast parallel algorithms and implementations on the Connection machine (and others) for a wide class of computational geometry problems, using sophisticated randomized sampling and load balancing techniques to improve the performance of the implementations.

These and other ongoing investigations are described below.

(1) Michael Landis (graduate student), John Reif (PI), and Robert Wagner (Duke faculty): Intermediate Representation for Parallel Implementation

Our research efforts are focused on the possibility of extending a high-level data-parallel language with constructs for process parallelism. Our goal is to begin with a data-parallel language like NESL, which is under development by Guy Blelloch at Carnegie Mellon University. This language provides nested data-parallelism. We believe that by extending it with process parallel primitives, the language will have wider applicability, but yet will still be able to be implemented efficiently.

We are focusing our efforts currently on the extension of a run-time library for implementing data-parallel languages. This library will provide the support for high-level language development while maintaining portability and efficiency through the use of the C language.
As an example, one possibility which we are investigating is the integration of the POSIX thread package with CVL, the C Vector Library under development at CMU.

A-1

(2) Michael Landis (graduate student), John Reif (PI), and Robert Wagner (Duke faculty): Data Movement on Processor Arrays

We have completed our study of developing ways of evaluating uniform expressions in near minimum parallel time on higher-dimensional processor arrays. A paper describing the solution on two-dimensional arrays is ready for submission to a journal publication. (This paper is a follow-up to Robert Wagner's paper, "Evaluating Uniform Expressions Within Two Steps of Minimum Parallel Time", Which solved the problem for two-dimensional arrays only.)

(3) John Reif (PI): Data-Parallel Implementations of Fast Multipole Algorithms for N-Body Interaction

Summary:

We are exploring data-parallel implementations of Fast Multipole Algorithms (FMA) for computing N-body interaction. Several algorithmic variants of FMA, such as adaptive FMA and other fastest known improvements [Reif,Tate92] are being expressed in a data-parallel fashion using the languages NESL (Nested Sequence Language, by Blelloch at CMU) and Proteus (at Duke and UNC). The data-parallel model provides a succinct high-level expression which exposes parallelism in a scalable fashion, and facilitates exploration and comparison of the parallel time complexity of algorithmic variants. Implementations are realized by transformation of the data-parallel programs to a lower-level widely portable vector model (VCODE), for example targeting the CM-5.

Details:

Many-body simulation is the key computational component in many challenging problems such as fluid mechanics and molecular dynamics simulation; the potential benefits of the latter include computer aided drug design and protein structure determination. In N-body simulation the goal is to simulate for a collection of N particles distributed in space the motion over time due to gravitational or electrostatic interaction between the particles. The naive solution requires N^2 comparisons to compute forces arising from pairwise interaction. More sophisticated algorithms reduce this complexity by relying on approximation of the lesser effects of far-away clusters of particles (perhaps modeling them by a few large particles), and on multigrid techniques which exploit this approximation by hierarchically decomposing the particle space into near and far-away points in order to isolate these far-field interactions.

The Fast Multipole Algorithm (FMA) [Greengard87] is a linear-time algorithm for calculating N-body interactions which uses multipole expansions to approximate the potential field created by a collection of bodies outside the region that contains the bodies. We have expressed an algorithmic variant in a data-parallel manner using the Proteus language. An abstract of a paper recently presented at DAGS'93 describing this effort follows.

A Data-Parallel Implementation of the Adaptive Fast Multipole Algorithm
by
Lars S. Nyland, Jan F. Prins, John H. Reif

Abstract

Given an ensemble of n bodies in space whose interaction is governed by a potential function, the N-body problem is to calculate the force on each body in the ensemble that results from its interaction with all other bodies. An efficient algorithm for this problem is critical in the simulation of molecular dynamics, turbulent fluid flow, intergalactic matter and other problems. The fast multipole algorithm (FMA) developed by Greengard approximates the solution with bounded error in time O(n). For non-uniform distributions of bodies, an adaptive variation of the algorithm is required to maintain this time complexity.

The parallel execution of the FMA poses complex implementation issues in the decomposition of the problem over processors to reduce communication. As a result the 3D Adaptive FMA has, to our knowledge, never been implemented on a scalable parallel computer. This paper describes several variations on the parallel adaptive 3D FMA algorithm that are expressed using the data-parallel subset of the high-level parallel prototyping language Proteus. These formulations have implicit parallelism that is executed sequentially using the current Proteus execution system to yield some insight into the performance of the variations. Efforts underway will make it possible to directly generate vector code from the formulations, rendering them executable on a broad class of parallel computers.

(4) Peter Mills (Research Associate) with John Reif: Rate Control in Parallel Algorithms

Summary:

Recent work has focused on extending high-level parallel computation paradigms with constructs for expressing relative rates of progress. The introduction of rate control supports a succinct specification of intended resource allocation, and is a first step in extending models of parallel computation with real-time properties, such as processor rates, in order to support timing analysis. We are currently pursuing implementation of the rate construct on a sequential interpreter for the Proteus language to use in experiments with algorithmic variations of adaptive N-body simulation.

Details:

A paper describing the rate construct and various applications appeared in the 1993 IEEE Workshop on Real-Time Parallel and Distributed Systems. An abstract of this paper follows.

Rate Control as a Language Construct for Parallel and Distributed Programming

by

Peter H. Mills, Jan F. Prins, and John H. Reif

Abstract

This paper introduces a new parallel programming language construct, the rate construct, and examines its utility for a variety of problems. The rate construct specifies constraints on the relative rates of progress of tasks executing in parallel, where progress is the amount of computational work as measured by elapsed ticks on a local logical clock. By prescribing expected work, the rate construct constrains the allocation of processor-time to tasks needed to achieve that work; in a parallel setting this constrains the distribution of tasks to processors and multiprocessing ratios, effected for example by load balancing. We present definitions of rate and underlying real-time primitives as orthogonal extensions to the architecture-independent parallel programming language Proteus. The utility of the rate construct is evidenced for a variety of problems, including weighted parallel search for a goal, adaptive many-body simulation in which rates abstract the requirements for load-balancing, and variable time-stepped computations in which the use of rates can alter the frequency of asynchronous iterations.

We are currently pursuing sequential implementation of the rate construct, and are also investigating means of transforming rate primitives in a parallel setting to lower-level real-time and scheduling constructs.

(5) Peter Mills (Research Associate) with John Reif: Implementing Asynchronous Parallelism using Tagged-Memory

Summary:

Recent efforts have concentrated on extending high-level parallel computation models with abstractions for asynchronous concurrency which roughly mimic tagged memory. A novel construct, guarded communication using linear operators, has been introduced and methods of extending parallel functional languages such as NESL (CMU) and Concurrent ML (Bell Labs) with linear operators are under investigation. A scalable extension for asynchronism in a functional style promises to have large impact in expressing and implementing parallel algorithms for machines such as CM-5 and KSR-1.

Detail:

We are developing high-level mechanisms for asynchronous concurrency which include a variant of synchronization variables and a novel construct we call linear variables. Synchronization variables are a synchronization mechanism found in coordination languages such as PCN and CC++ as well as in Id's I-structures. Linear variables are a further extension which model resource consumption, and prove valuable in succinctly modeling channel and rendezvous operations within a shared-memory framework. Linear variables prove particularly advantageous in that they can be readily ported to many architectures, and promise to be amenable to optimization techniques which transform the program to decrease non-local references.

We are investigating extending an existing widely portable data-parallel language, CMU's NESL (supporting nested data parallelism) with a wrapper for asynchronous parallelism built on linear variables (similar to Id's M-structures). The intent is to extend and thus capitalize on existing techniques for transforming nested data parallelism to vector models, i.e. the transformation of NESL to VCODE. (Such an implementation strategy will most likely rely on run-time library extensions rather than extensions to a low-level intermediate representation).

(6) Peter Su (postdoc) and John Reif: Implementations of Parallel Algorithms in Computational Geometry

With Peter Su, a graduate student from Dartmouth working at Duke on his Ph.D. on parallel algorithm implementations with Reif, we are investigating parallel algorithms for constructing Voronoi Diagrams and related problems in computational geometry. Our interest is not only to build effective algorithms for these problems, but also to consider the kinds of tools that make such work easier and more effective.

Su recently defended his dissertation at Dartmouth in June 1993. An abstract of Su's dissertation follows.

Efficient parallel algorithms for closest point problems

by

Peter Su

Abstract

This dissertation develops fast algorithms for solving closest point problems on parallel and vector computers. Algorithms for such problems have applications in many areas including statistical classification, crystallography, data compression, and finite element analysis. We present a simple and flexible programming model for designing and analyzing parallel algorithms. Also, fast parallel algorithms for nearest-neighbor searching and constructing Voronoi diagrams are described. Finally, we demonstrate that the algorithms actually obtain good performance on a wide variety of machine architectures, including the MasPar MP-1, Cray Y-MP and KSR-1 supercomputers.

The key algorithmic ideas that used to obtain good performance are exploiting spatial locality, and random sampling. Spatial decomposition provides allows many concurrent threads to work independently of one another in local areas of a shared data structure. Random sampling provides a simple way to adaptively decompose irregular problems, and to balance workload among many threads. Used together, these techniques result in effective algorithms for a wide range of geometric problems.

The key experimental ideas used in this research are simulation and animation. Algorithm animation is used to validate algorithms and gain intuition about their behavior. The expected performance of algorithms is modeled using simulation experiments and some knowledge as to how much critical primitive operations will cost on a given machine. In addition, this is done without the burden of esoteric computational models that attempt to cover every possible variable in the design of a computer system. An iterative process of design, validation, and simulation delays the actual implementation until as many details as possible are accounted for. Then, further experiments are used to tune implementations for better performance.

(7) Shenfeng Chen with John Reif: Parallel Sort Implementation

Summary:

The fastest known sort is a parallel implementation of radix sort in a CRAY, due to CMU's Guy Blelloch. The current sorting algorithms on parallel machines like the Cray and CM-2 use radix and bucket sort. But they do not take advantage of the possible distribution of the input keys. We are developing an algorithm using data compression to achieve a fast parallel algorithm which takes this advantage. We expect the new algorithm to beat the previous fastest sort by a few factors. We are working to implement this new parallel sorting algorithm on various parallel machines. A paper describing our recent efforts "Using Learning and Difficulty of Prediction to Decrease Computation: A Fast Sort and Priority Queue on Entropy Bounded Inputs", has been accepted to appear in FOCS'93.

Details:

Radix sort is very efficient when the input keys can be viewed as bits. But the basic radix sort is not distribution-based so it needs to look up all digits.

Our approach is to find the structure (distribution) of the input. This is achieved by sampling from the original set. Then a hash table is build from those sample keys. All keys are indexed to buckets separated by consecutive sample keys. A probability analysis shows that the largest set can be bounded within a constant of the average size.

The indexing step is made faster by binary searching the hash table for match. From previous result, each hash function computation needs only constant time.

Our algorithm needs O(nloglogn) sequential time given that the compression ratio of the given input set is not too big. In parallel, our algorithm works well in chain-sorting. In list ranking sorting, the total work is also reduced.

We have implemented this algorithm on Sparc II and compared its performance with the system routine quicksort. It turns out that our algorithm outwins the quicksort() for sufficiently large number of keys (32M). Thus, it may find its place in sorting large database operations (e.g., required by joint operations). In these applications the keys are many words long so our algorithm is even more advantageous in this case where the cutoff is much lower.

Also we implemented this algorithm on the Cray Y-MP using one processor. The result is similar to that for the Sparc II.

We also give some applications of our algorithm to computational geometry problems: 2-D convex hull and trapezoidal decomposition assuming that the input are entropy bounded.

(8) Deganit Armon (A.B.D.) with John Reif: Dynamic Graph Separator Algorithms.

Summary:
We continued work on dynamic graph problems, using the techniques we developed when studying the dynamic separator problem. These are techniques for converting a fixed input randomized algorithm into a randomized algorithm that accepts changes to the input. In addition we showed a method for converting an expected time randomized algorithms to randomized algorithms with high likelihood time bounds. We attempted to apply these techniques to other dynamic graph problems, in particular dynamic nested dissection and planar graph algorithms.

Details:
Randomized algorithms that use sampling select a small sample of the input, apply an "expensive" algorithm to the sample, and then extrapolate the result to the entire dataset. The solution will not necessarily be "exact", but the error can usually be bounded. Examples of such algorithms range from the version of quicksort in which a pivot is found by taking the mean of a small sample, to complex algorithms for finding graph separators, to implementations in computational geometry. We described a technique for transforming such algorithms so that they can deal with dynamically changing input, and applied this method to the problem of finding a sphere separator for a set of points. We showed that while the static algorithm takes linear time, computing a separator after adding or deleting a point from the input set requires only a logarithmic number of steps. We also showed that maintaining a more complex separator structure could also be done dynamically in polylog time.

Another characteristic of randomized algorithms is that while we can determine the expected time to completion, the actual running times may vary considerably. We showed a technique which, through the use of multiple processes (called replicants) which are performing the same computations, we can guarantee the expected time bounds (with some slowdown) with high likelihood. This technique is particularly useful when in addition to changing the input the algorithm is also presented with queries about the input. We can thus guarantee timely processing of a query by one or more of the replicants. We showed how this method can be applied to the problem of maintaining graph separators with only a log^2 slowdown. This method can be applied to other randomized algorithms that involve maintaining a data structure and answering queries, such as arise in computational geometry.

A paper describing these techniques and their application to the dynamic sphere separator problem has been submitted to WADS 93. Currently we are working on finding randomized algorithms which can be dynamized using these techniques.

(9) Prokash Sinha with John Reif: Randomized Parallel Algorithms for Min Cost Paths

Summary:
We have completed our initial investigation to derive randomized parallel algorithms for Min Cost Paths in a Graph of High Diameter. Our present accomplishment is a randomized sequential algorithm with an order of magnitude performance gain for some dense graphs.

We also found a similar result for PRAM computational model which meets the work we proposed to do in our paper "A Randomized Algorithm for Min Cost Paths in a Graph of High Diameter: Extended Abstract" (J. Reif and P. Sinha). Currently we are in the process of submitting our findings to technical journals and conferences. Our next phase of work would include similar derivations of randomized parallel algorithms for a wide variety of discrete structures which arises naturally in the area of Graph Theory and Combinatorics. Our current research effort is to extend the techniques of Flajolet and Karp to develop techniques and tools for timing analysis of algorithms. This effort is to derive tools for semiautomatic randomized analysis.

(10) Hongyan Wang with John Reif: Control of a VLSR System with Distributed Control Mechanism

Summary
In our previous work, we proposed a molecular dynamics approach for distributed control of Very Large Scale Robotics (VLSR) system. We showed that a system of large number of robots can stabilize to certain

patterns under given force functions. We call this level of control the lower level control of the system. We further study the high level control. The high level control problem is that given a desired distribution pattern, how we can choose appropriate force functions (i.e. determine the coefficients in force functions) to achieve the pattern.

Details

In our previous work ("Social Potential Fields: A Molecular Dynamics Approach for Distributed Control of Multiple Robots" [J. Reif, H. Wang]), we proposed a molecular dynamics approach for distributed control of VLSR. We view our VLSR systems as a molecular dynamics system, with predefined force laws between each ordered pair of components (robots, obstacles, objectives and other configurations). However these laws may differ from molecular systems in that we allow the controller to arbitrarily define distinct laws of attraction and repulsion for separate pairs and groups of robots to reflect their socia' relations or to achieve some goals. For example, we define a pair-wise force law of repulsion and attraction for a group of identical robots. The repulsion will prevent collision among robots and the attraction will keep them in a cluster. Once the force laws are set up (they can be modified by the global controller), each individual's movement is computed locally according to the local environment sensed by individual robots and the force laws.

We did computer simulations involving large numbers of robots. These simulations show that for chosen control parameters (coefficients in the force functions), the system can stabilize to certain desired patterns, e.g. forming a more or less evenly distributed single cluster, simulating attacking and guarding strategies. The force functions used in the simulations are defined intuitively to reflect the relations of different groups. Now we are searching for a systematic way of computing the coefficients for the force functions to achieve a certain pattern.

In later work ("A Constant Time Algorithm for N-body Simulation with Smooth Distributions" [J. Reif, H. Wang]), we proposed to use density function to describe the distribution of large number of robots in our VLSR system and proposed a constant time algorithm to compute the density function. Let C denote the vector of coefficients in the force functions and we call C the control vector. The density function D(x,y) is computed for a given control vector C. The boundary of the distribution is an implicit function as D(x,y)=u, where u is a threshold. D(x,y)=0 if D(x,y)<u. Since D is a smooth function, we want to put a cut-off u such that the integral of D(x,y) of the area where D(x,y)>u equals to N, the number of robots.

The control problem can be stated as given a density function or a boundary function, find the correct control vector C, so the desired density function or boundary function can be approximated.

We can consider D(x,y) as an implicit function of also the vector C. The problem of achieving a good approximation is a problem of minimizing the function: integral of ($D(x,y)-D^*(x,y))^2$,
where D^* is the desired density function. Let the function be denoted H. We want to solve the equation $dH/dC = 0$. Since H is not an explicit function of C, we use Quasi Newton Method to solve this equation.
Similarly for the control of boundary function of the distribution.

Thus given a desired distribution pattern, the global control can compute the appropriate control vector C and broadcast the vector to the system of robots. Each robot will update their table of force functions accordingly. The motion is still decided by individual robots locally, but using the new force functions.

Our work will also be extended to 3-dimensional cases.

(11) Hongyan Wang with John Reif: On Line Navigation Through Regions of Variable Densities.

Summary
Most of the previous work on on-line navigation focused on the problem of navigating through an unknown terrain with impenetrable obstacles. It is interesting and practical to consider on-line navigation problems where the obstacles are penetrable. Consider a robot traveling in a field to some target. Lakes, swamps and hills can be considered as obstacles that are penetrable, but require more effort per unit length on penetrating. Some competitive on-line algorithms for impenetrable obstacles are no longer competitive for the above scenario with respect to the effort consumed traveling along the path.

Details
The general model of the problem is as follows. Each obstacle is a polygon with a homogeneous density. The density of an obstacle is the effort required to travel a unit length through the obstacle. We normalize the density of free space to 1 and the densities of any obstacles should be no less than 1. The density of each obstacle is unknown to the robot until the robot touches the obstacle. The robot is considered as a point object and can use only tactile information.

The competitive ratio is the worst case ratio of the effort to travel along the path computed by the on-line algorithm to the least effort needed to get the the target.

In [Blum, Rahhavan, Schieber91] two kinds of problems are defined as the wall problem, where the target is an infinite line and the obstacles are oriented rectangles, and the room problem, where the obstacles are oriented rectangles that are confined to lie within a square "room", and the target is a point in the room. In all the problems, the robot can only use tactile information. For the wall problem, Blum et al. gave an algorithm that achieves an upper bound of O(n^(1/2)) on the ratio, matching the lower bound given in [Papadimitriou, Yannakakis89], where n is the Euclidean distance from the source point to the target line. This algorithm is not competitive if the obstacles are penetrable, for example consider the scenario where the obstacle is very thin but very long. Their algorithm uses so called sweeping strategy.

First we studied the Wall Problem with Penetrable Obstacles, where each rectangular obstacle has a homogeneous density. We showed that the optimal competitive ratio of O(n^(1/2)) can still be achieved with some modification to the original sweeping algorithm presented in [Blum, Rahhavan, Schieber91].

Then we generalized the Wall Problem to allow obstacles with higher densities within an obstacle. We call this problem the Recursive Wall Problem. Now finding a path through an obstacle can be considered as a Recursive Wall Problem as well. A lower bound of competitive ratio is shown to be Omega(N^(1/2)), where N = n_0n_1...n_(k-1). k is the level of recursion of the problem and n_i is the upper bound of expanded Euclidean distances of obstacle of level i. Recursively applying the sweeping strategy, we showed that the lower bound can be achieved. Thus we gave an optimal algorithm for the Recursive Wall Problem.

(12) Akitoshi Yoshida with John Reif: Image and Video Compression

We considered several compression techniques using optical systems. Optics can offer an alternative approach to overcome the limitations of current compression schemes. We gave a simple optical system for the cosine transform. We designed a new optical vector quantizer system using holographic associative matching and discussed the issues concerning the system.

Optical computing has recently become a very active research field. The advantage of optics is its capability of providing highly parallel operations in a three dimensional space. Image compression suffers from large computational requirements. We propose optical architectures to execute various image compression techniques, utilizing the inherent massive parallelism of optics.

In our paper[RY2], we optically implemented the following compression and corresponding decompression techniques:

o transform coding
o vector quantization
o interframe coding for video

We showed many generally used transform coding methods, for example, the cosine transform, can be implemented by a simple optical system. The transform coding can be carried out in constant time.

Most of this paper is concerned with an innovative optical system for vector quantization using holographic associative matching. Limitations of conventional vector quantization schemes are caused by a large number of sequential searches through a large vector space. Holographic associative matching provided by multiple exposure holograms can offer advantageous techniques for vector quantization based compression schemes. Photo-refractive crystals, which provide high density recording in real time, are used as our holographic media. The reconstruction alphabet can be dynamically constructed through training or stored in the photorefractive crystal in advance. Encoding a new vector can be carried out by holographic associative matching in constant time.

We also discussed an extension of this optical system to interframe coding.

<u>On going work:</u>
We are investigating optical algorithms for video compression.

(1) Computational Geometry by Optical Computers
Some problems require inherently high degrees of interconnections which may not be provided by any conventional electrical computers. The advantage of optical computers is their apparent parallelism in a three dimensional space. Several computational models have been already proposed and constructed by various research groups. As the progress of optical computers continues, there is a great demand in designing and investigating various algorithms that are efficient and appropriate for the proposed models. This situation resembles to the one a decade ago, when

various algorithms were investigated for the theoretical VLSI model. Thus, we understand that the investigation on optical computing algorithms will be essential to the development of optical or hybrid massively parallel computers.

Optical techniques are particularly suited for processing images. This leads us to believe that many problems found in computational geometry may be efficiently solved by optical computers. Some researchers have recently started to investigate some basic problems. We have been investigating these and some other problems. We have obtained some new results.

(2) Optical Interconnection
Among processing units placed on a plane, various space-invariant interconnections can be holographically established in constant time. We are investigating appropriate interconnections and efficient algorithms for several problems.

(3) Efficient computation for optical scattering
An efficient algorithm to solve the Helmholtz equations was developed by Rokhlin at Yale. We have been studying his algorithm.

(4) Simulation of optical computing algorithms
We implemented a software simulator for optical computing algorithms. The simulator is written in C on the X-window environment. It has a lisp-like user interface, and images, which are the basic data structures in the optical computing algorithms, are treated as lisp objects. We simulated some algorithms designed for computational geometry problems.

We are improving the simulator and planning to implement it on a parallel machine.

(13) Researchers supported (other than PI):

Mike Landis, graduate student
Peter Mills, post-doc
Peter Su, visiting graduate student
Robert Wagner, professor
Akitoshi Yoshida, graduate student

(14) Publications

(1) *Efficient Parallel Algorithms For Closest Point Problems*, Peter Su. Ph.D. Dissertation, Dartmouth College, Hanover, NH, June 1993.

(2) *Parallel Algorithm Derivation and Program Transformation* (R. Paige, J. Reif, and R. Wachter, eds.). Morgan-Kaufmann Publishers, 1993.

(3) *Syn'hesis of Parallel Algorithms* (J. Reif, ed.). Kluwer Academic Pu' ishers, 1993.

(4) "A Dynamic Separator Algorithm", D. Armon and J. Reif. To appear in *Workshop on Algorithms and Data Structures (WADS'93)*, Montreal Canada, Aug. 1993.

(5) "Parallel and Output Sensitive Algorithms for Combinatorial and Linear Algebra Problems", J. Cheriyan and J. Reif. To appear in *Symposium on Parallel Algorithms and Architectures (SPAA'93)*, Velon, Germany, July 1993.

(6) "Using Learning and Difficulty of Prediction to Decrease Computation: A Fast Sort and Priority Queue on Entropy Bounded Inputs", S. Chen and J. Reif. To appear in *Symposium on Foundations of Computer Science (FOCS'93)*, Palo Alto, CA, Nov 1993.

(7) "The Complexity of N-body Simulation", J. Reif and S. Tate. To appear in *20th International Colloquium on Automata, Languages, and Programming (ICALP'93)*, Lund, Sweden, July 1993.

(8) "An O(nlog^{3}n) Algorithm for the Real Root Problem with Application to the Symmetric Eigenvalue Problem", J. Reif. To appear in *Symposium on Foundations of Computer Science (FOCS'93)*, Palo Alto, CA, Nov 1993.

(9) R. Paturi, J. Reif and S. Rajasekaran, "The Light Bulb Problem," to appear in *Information and Computation*, 1993. Also appeared as "Efficient and Robust Learning Using Statistical Bootstrap," in *Proc. Workshop on Computational Learning Theory*, August 1989.

(10) "Optical Techniques for Image Compression", J. Reif and A. Yoshida. *Proc. 2nd Annual IEEE Data Compression Conference (DCC 92)*, Snowbird, UT, March 1992. Also to appear in *Image and Text Compression* (J.A. Storer, ed.), Kluwer Academic Publishers.

(11) "Rate Control as a Language Construct for Parallel and Distributed Programming", P. Mills, J. Prins, and J. Reif. *Proc. of IEEE Workshop on Parallel and Distributed Real-Time Systems (IPPS'93)*, Newport Beach, CA, April 1993.

(12) "Memory-Shared Parallel Architectures for Vector Quantization Algorithms", T. Markas and J. Reif. *Picture Coding Symposium*, Lusanne Switzerland, March 93. Submitted for journal publication.

(13) "Multispectral Image Compression Algorithms", T. Markas and J. Reif. *Data Compression Conference (DCC'93)*, Snowbird, Utah, March 1993.

(14) "A Data-Parallel Implementation of the Adaptive Fast Multipole Algorithm", L. Nyland, J. Prins, and J. Reif. *Proc. 2nd Symposium on Issues and Obstacles in the Practical Implementation of Parallel Algorithms and the Use of Parallel Machines (DAGS'93)*, Hanover, N.H., June 1993.

(15) "Quad Tree Structures for Image Compression Applications", T. Markas and J. Reif. To appear in special issue of *Journal of Information Processing and Management*, 1992.

(16) "Continuous Alternation: The Complexity of Pursuit in Continuous Domains", J. Reif and S. Tate. To appear in a special issue of *Journal of Algorithmica* (B. Donald, ed.), 1992.

(17) "Searching in an Unknown Environment", M. Kao, J. Reif, and S. Tate. To appear in *Information and Computation*. Also appeared in *Proc. 4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA92)*, San Diego, CA, 1992.

(18) "Diffraction Realization Of An Optical Expander", R. Barakat and J. Reif. Submitted for publication, 1993.

(19) "On Parallel Implementations and Experimentations of Lossless Data Compression Algorithms", T. Markas and J. Reif. Submitted for publication, 1992.

(20) "A Constant Time Algorithm for N-body Simulation with Smooth Distributions", J. Reif and H. Wang. Submitted for publication, 1992.

(21) "Reduction on Multidimensional Processor Arrays", R. Wagner and M. Landis. Submitted for publication, 1993.

(22) "Improving viewing condition for reduced information holographic display", J. Reif and A. Yoshida. Submitted for publication, 1993.

(23) "Dynamic Algebraic Algorithms", J. Reif and S. Tate). Submitted for publication, 1992.

(24) "Dynamic Parallel Tree Contraction", J. Reif and S. Tate. Submitted for publication, 1992.

(25) "Social Potential Fields: A Molecular Dynamics Approach for Distributed Control of Multiple Robots", J. Reif and H. Wang. Submitted for publication, 1992.

(26) "A Randomized Algorithm for Min Cost Paths in a Graph of High Diameter: Extended Abstract", J. Reif and P. Sinha. Submitted for publication, 1992.

(27) "A Method for Deriving Systolic Algorithms", R. Wagner and M. Landis. Submitted for journal publication, 1992.

(28) "Evaluating Uniform Expressions Within Two Steps of Minimum Parallel Time", R. Wagner. Submitted for journal publication, 1992.